ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Образовательная программа бакалавриата «Программная инженерия»

СОГЛАСОВАНО Научный руководитель, профессор департамента программной инженерии факультета компьютерных наук канд. техн. наук	УТВЕРЖДАЮ Академический руководитель образовательной программы «Программная инженерия» профессор департамента программной инженерии, канд. техн. наук
_____ С.М. Авдошин «___» _____ 2020 г.	_____ В.В. Шилов «___» _____ 2020 г.

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

ПРОГРАММА КЛАССИФИКАЦИИ КОМПЬЮТЕРНЫХ АТАК ПО НАБОРУ ДАННЫХ ISCX BOTNET

Техническое задание

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.02.13-01 ТЗ 01-1-ЛУ

Исполнитель
студент группы БПИ193
_____ /Е.А. Гриценко /
«____»_____ 2020 г.

Москва 2020

УТВЕРЖДЕН
RU.17701729.02.13 -01 ТЗ 01-1-ЛУ

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

ПРОГРАММА КЛАССИФИКАЦИИ КОМПЬЮТЕРНЫХ АТАК ПО НАБОРУ ДАННЫХ ISCX BOTNET

Техническое задание

RU.17701729.02.13-01 ТЗ 01-1

Листов 18

Москва 2020

ОГЛАВЛЕНИЕ

1. Введение....4
1.1. Наименование программы....4
1.2. Краткая характеристика области применения....4
2. Основания для разработки....5
2.1. Документ, на основании которого ведётся разработка....5
2.2. Наименование темы разработки....5
3. Назначение разработки....6
3.1. Функциональное назначение....6
3.2. Эксплуатационное назначение....6
4. Требования к программе....7
4.1. Требования к функциональным характеристикам....7
4.1.1. Требования к составу выполняемых функций....7
4.1.2. Форматы входных и выходных данных....8
4.2. Требования к надёжности....9
4.3. Условия эксплуатации....9
4.4. Требования к составу и параметрам технических средств....10
4.6. Требования к маркировке и упаковке....10
4.8. Специальные требования....10
5. Требования к программной документации....12
5.1. Состав программной документации....12
5.2. Специальные требования к программной документации....12
6. Технико-экономические показатели....13
6.1. Ориентировочная экономическая эффективность....13
6.2. Предполагаемая потребность....13
6.3. Экономические преимущества по сравнению с отечественными и зарубежными аналогами....13
7. Стадии и этапы разработки....14
7.1. Стадии разработки....14
7.2. Сроки разработки и исполнители....15

8. Порядок контроля и приёмки.....16
9. Список использованных источников.....17
Лист регистрации изменений.....18

1. ВВЕДЕНИЕ

1.1. Наименование программы

ПО состоит из двух чатей: библиотеки классов и програмы. Наименование программы — “BotnetDetector”, наименование библиотеки — “LibBtntDtct”. Общее наименование ПО — “BotnetDetector”.

1.2. Краткая характеристика области применения

Областью применения программы “BotnetDetector” является информационная безопасность. Под botnet-ом понимается сеть компьютерных устройств, в частности — IoT-устройств, заражённых вирусом (bot-ом), который предоставляет атакующему возможность удалённого массового выполнения на них команд и удалённого использования: в частности — для проведения DDOS атак, спам-рассылок, mining-а криптовалют.

Программа призвана помочь обеспечить защиту от botnet-ов, путём их обнаружения в сетевом трафике.

2. ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

2.1. Документ, на основании которого ведётся разработка

Документом, на основании которого ведётся разработка, является приказ декана факультета компьютерных наук И.В. Аржанцева „Об утверждении тем, руководителей курсовых работ студентов образовательной программы «Программная инженерия» факультета компьютерных наук“ № 2.3-02/1112-04 от 11.12.2019.

2.2. Наименование темы разработки

Наименование темы разработки — «Программа классификации компьютерных атак по набору данных ISCX Botnet» (“ISCX Botnet Dataset Classification Attack Program”).

3. НАЗНАЧЕНИЕ РАЗРАБОТКИ

3.1. Функциональное назначение

Программа предназначена для классификации сетевых потоков в файле с информацией о них, трафике с доступных сетевых устройств или его дампе как порождённых ботнетом или же обычных потоков, с возможностью дополнения и использования с иными пользовательскими классификаторам, в частности, небинарными без рекомпиляции (plug-in). Также программа предназначена для установления точности классификаторов путём их выполнения на файлах, содержащих информацию о потоках и именах их классов.

3.2. Эксплуатационное назначение

Программа призвана помочь обеспечить защиту от botnet-ов, путём классификации сетевых потоков как потоков, порождённых botnet-ом, или же как обычных потоков, а также помочь в тестировании классификаторов, предоставив платформу для их выполнения на трафике с доступных сетевых устройств, его дампах или файлах, содержащих информацию о потоках.

4. ТРЕБОВАНИЯ К ПРОГРАММЕ

4.1. Требования к функциональным характеристикам

4.1.1. Требования к составу выполняемых функций

4.1.1.1. Наличие библиотеки классов .NET, которая:

4.1.1.1.1. Позволяет производить захват информации о потоках из трафика с доступных сетевых устройств или дампа трафика в формате PCAP.

4.1.1.1.2. Позволяет представлять информацию о потоке в строковом виде и его чтение (в формате 4.1.2.1).

4.1.1.1.3. Позволяет производить потоковую классификацию сетевых потоков, представленных в виде экземпляров классов библиотеки: как «обычных» или же как порождённых botnet-ом.

4.1.1.1.4. Содержит набор базовых классов для пользовательских классификаторов. Предоставляет возможность определения пользователем небинарных классификаторов. Имена классов не должны содержать пробелы в начале и в конце, а также запятые.

4.1.1.1.5. Содержит механизм, позволяющий производить классификацию потоков из трафика сетевых устройств или PCAP-файла по мере захвата информации о них.

4.1.1.2. Наличие консольного .NET приложения, предоставляющего интерфейс взаимодействия с библиотекой, которое:

4.1.1.2.1. Позволяет, используя библиотеку, открывать файл дампа сетевого трафика в формате PCAP, определять в нём потоки и производить классификацию потоков в них. Позволяет по мере чтения файла отображать статистику о прочитанных пакетах и классифицированных потоках.

4.1.1.2.2. Позволяет, используя библиотеку, определять в трафике, получаемом с видимого ОС сетевого устройста, потоки и производить их классификацию в реальном времени. Позволяет отображать статистику об обработанных и находящихся в очереди пакетах, пропущенных (в силу перегруженности) пакетах и классифицированных потоках.

4.1.1.2.3. Позволяет, используя библиотеку, классифицировать потоки, получая информацию о них из файла формата, описанного в пункте 4.1.2.3. Позволяет по мере чтения файла отображать статистику о прочитанных пакетах и классифицированных потоках.

4.1.1.2.4. Для всех трёх вышеприведённых действий:

- Позволяет записывать информацию о потоках или определённых классах в файл или стандартный вывод, таким образом обеспечивая возможность дальнейшей обработки классифицированных потоков иными программами. Запись осуществляется в одном из CSV форматов, описанных в пунктах 3 — 5 главы 4.1.2.
- Позволяет выбирать для записи только те потоки или классы, класс которых отличен от нормального.
- В качестве классификатора предоставляет возможность выбрать как библиотечный бинарный классификатор, так и сторонние пользовательские классификаторы.
- Позволяет производить бинарную классификацию потоков трафика как нормальных, или как ботнет-потоков. Позволяет производить классификацию пользовательскими классификаторами, в частности, небинарными, и на их основании проводить уже бинарную классификацию, полагая, что все потоки, не попадающие в класс нормальных, являются ботнет-потоками.

4.1.1.2.5. Позволяет по файлу, содержащему информацию о потоках и их классах формата, описанного в пункте 4.1.2.4, производить вычисление точности классификатора и сравнение полученных им результатов — вычисленных классов — с указанными в файле.

4.1.2. Форматы входных и выходных данных

4.1.2.1. Формат строкового представления информации о потоке: «start, end, source, sport, destination, dport, protocolID, packetsCount, outgoingPackets, octetsCount, payloadOctetsCount». Где «start» и «end» — даты первого и последнего пакета в данном потоке в формате «dd.MM.yyyy HH:mm:ss.ffffff», где «dd» — номер дня месяца, «MM» — номер месяца, «yyyy» — номер года, «HH» — часы от 00 до 23 включительно, «mm» — минуты, «ss» — секунды, «ffffff» — микросекунды. Незначащие нули должны сохраняться для корректного чтения. «source» и «destination» — IP-адреса источника и назначения в представлении .NET

Framework 4.7. «sport» и «dport» — порты источника и назначения (числа от 0 до 65535 включительно). «prototcolID», «packetsCount», «outgoingPackets» — номер протокола, число пакетов и число исходящих пакетов — десятичные числа, представимые в типе int C#. «octetsCount», «payloadOctetsCount» — общее количество октет и количество октет в payload-е пакетов — десятичные числа, представимые в типе long C#. При чтении допускается наличие после значения «payloadOctetsCount» иных символов, если первый из них — запятая. При чтении также допускается наличие обычных пробелов на конце строкового представления.

4.1.2.2. Формат строкового представления классированного представления информации о потоке: «flowInfo, classLabel». Где «flowInfo»— строковое представление информации о потоке (4.1.2.1), а «classLabel» — название класса, присуждаемого данному потоку. Таким образом, формат совместим на чтение с форматом строкового представления информации о потоке.

4.1.2.3. Формат файла или потока с информацией о потоках. Каждая строка является строкой, соответствующей формату информации о потоке. При этом при чтении допускается отличие первой строки от этого формата, не являющейся информацией о потоке, а также наличие пустой строки в конце (символа переноса).

4.1.2.4. Формат файла или потока с классированной информацией о потоках. Каждая строка файла является строкой, соответствующей формату классированного информации о потоке. При этом при чтении допускается отличие первой строки от этого формата, не являющейся информацией о потоке, а также наличие пустой строки в конце. Таким образом этот формат соответствует формату 4.1.2.3.

4.1.2.5. Формат файла или потока с информацией о классах. Каждая строка файла или потока представляет собой имя некоторого класса. Используется только для записи.

Вышеописанные форматы, таким образом, являются при записи CSV (comma separated values) форматами. При чтении и записи чисел используется инвариантная культура.

4.2. Требования к надёжности

Программа не должна завершаться аварийно при любых действиях пользователя в нёй (если не подразумевается отладка приложения).

4.3. Условия эксплуатации

Не требует специального обслуживания. Требуемая квалификация — продвинутый пользователь, имеющий навыки работы с консольными приложениями через терминал

ОС, обладающий минимальными знаниями в информационной безопасности и сетевых коммуникациях.

4.4. Требования к составу и параметрам технических средств

Процессор устройства должен обладать архитектурой, поддерживаемой реализациями .NET Framework 4.7 и библиотекой SharpPcap (например, x86-64 или aarch64). Для захваты и классификации потоков сетевого трафика в реальном времени устройство должно обладать сетевой картой, видимой операционной системой и библиотекой SharpPcap. Кроме того, устройство должно обладать достаточными вычислительными мощностями для проведения этой операции.

4.5. Требования к информационной и програмной совместимости

Программа должна быть написана на языке C# 7.0.

Для работоспособности программы необходимо наличие наличие на операционной системе корректно работающей реализации .NET Framework 4.7. Также необходима корректная работа библиотеки SharpPcap, поставляемой вместе с программой, для чего необходимо, как минимум, наличие её зависимостей (libpcap для GNU/Linux, npcap для Windows). Для анализа трафика в реальном времени необходимо, чтобы приложение обладало соответствующими разрешениями, позволяющими перехватывать трафик выбранного устройства, а также, чтобы система предоставляла приложению соответствующие скорости трафика вычислительные мощности. ОС должна быть способна предоставить как минимум 256 MB оперативной памяти. Сепаратор строк, используемый ОС, должен быть либо LF, либо CRLF.

4.6. Требования к маркировке и упаковке

Особых требований нет.

4.7. Требования к транспортированию и хранению

Программное изделие может храниться на жёстком диске и флеш-памяти. Транспортировка допускается посредством жёстких дисков, флеш-носителей, CD и DVD дисков, сети «Интернет».

4.8. Специальные требования

Программа должна быть разработана на основе тренировочной части набора данных “ISCX Botnet 2014” [1]. Точность классификатора и работоспособоность

программы должна быть в обязательном порядке протестирована на основе тестовой части этого набора данных.

5. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

5.1. Состав программной документации

- «Программа классификации компьютерных атак по набору данных ISCX Botnet». Техническое задание (ГОСТ 19.201-78);

- «Программа классификации компьютерных атак по набору данных ISCX Botnet». Программа и методика испытаний (ГОСТ 19.301-78);

- «Программа классификации компьютерных атак по набору данных ISCX Botnet». Текст программы (ГОСТ 19.401-78);

- «Программа классификации компьютерных атак по набору данных ISCX Botnet». Пояснительная записка (ГОСТ 19.404-79);

- «Программа классификации компьютерных атак по набору данных ISCX Botnet». Руководство оператора (ГОСТ 19.505-79);

- «Программа классификации компьютерных атак по набору данных ISCX Botnet». Руководство программиста (ГОСТ 19.504-79).

5.2. Специальные требования к программной документации

Документы к программе должны быть выполнены в соответствии с ГОСТ 19.106-78 и ГОСТами к каждому виду документа (см. п. 5.1.). Пояснительная записка должна быть загружена в систему Антиплагиат через LMS «НИУ ВШЭ». Лист, подтверждающий загрузку пояснительной записки, сдаётся в учебный офис вместе со всеми материалами не позже, чем за день до защиты курсовой работы. Техническое задание и пояснительная записка, титульные листы других документов должны быть напечатаны, подписаны академическим руководителем образовательной программы 09.03.04 «Программная инженерия», руководителем разработки и исполнителем перед сдачей курсовой работы в учебный офис не позже одного дня до защиты. Документация и программа также сдается в электронном виде в формате .pdf или .docx, в архиве формата .zip или .rar. За один день до защиты комиссии все материалы курсового проекта: техническая документация, программный проект, исполняемый файл, отзыв руководителя должны быть загружены одним или несколькими архивами в проект дисциплины «Курсовой проект 2019-2020» в личном кабинете в информационной образовательной среде LMS (Learning Management System) НИУ ВШЭ.

6. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

6.1. Ориентировочная экономическая эффективность

В рамках данной работы расчёт экономической эффективности не предусмотрен.

6.2. Предполагаемая потребность

Поскольку botnet-ы негативно влияют на работу устройств в них участвующих и используются для проведения с заражённых систем различных атак (DDOS, спам-рассылки), их наличие является крайне нежелательным. Разрабатываемое приложение призвано обнаруживать их следы в трафике, путём классификации сетевых потоков, что предоставляет пользователю инструмент защиты от botnet-ов.

6.3. Экономические преимущества по сравнению с отечественными и зарубежными аналогами

Анализ экономических преимуществ по сравнению с аналогами не производился.

7. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

7.1. Стадии разработки

I. Техническое задание

1. Обоснование необходимости разработки

- Постановка задачи;
- Сбор исходных материалов.

2. Научно-исследовательские работы;

- Определение структуры входных и выходных данных;
- Предварительный выбор методов решения задач;
- Обоснование целесообразности применения ранеее разработанных программ;
- Определение требований к техническим средствам;
- Обоснование принципиальной возможности решения поставленной задачи.

3. Разработка и утверждение технического задания

- Определение требований к программе;
- Определение стадий, этапов и сроков разработки программы и документации на неё;
- Выбор языков программирования;
- Определение необходимости проведения научно-исследовательских работ на последующих стадиях;
- Согласование и утверждение технического задания.

II. Рабочий проект

1. Подготовка к разработке программы

- Уточнение структуры входных и выходных даннных;
- Разработка алгоритмов и методов решения задачи и подзадач;
- Определение формы представления входных и выходных данных;
- Разработка структуры программы;
- Разработка пояснительной записки (ГОСТ 19.404-79).

2. Разработка программы

- Программирование и отладка программы.

3. Разработка програмной документации

- Разработка программной документации в соответствии с требованиями ГОСТ 19 ЕСПД (единой системы программной документации).

4. Испытания программы

- Разработка, согласование и утверждение программы и методики испытаний;
- Проведение испытаний программы в соответствии с утверждённой программой и методикой;
- Корректировка программы и программной документации по результатам испытаний.

III. Внедрение

1. Подготовка и передача программы

- Утверждение даты защиты программного продукта;
- Подготовка программы и программной документации для презентации и защиты;
- Представление разработанного програмного продукта руководителю и получение отзыва;
- Загрузка пояснительной записки в систему Антиплагиат через LMS НИУ ВШЭ;
- Загрузка материалов курсового проекта (курсовой работы) в LMS, проект дисциплины «Курсовой проект 2019-2020» (п. 5.2).

7.2. Сроки разработки и исполнители

Разработка должна закончиться к 15 мая 2020 года.

Исполнитель: **Гриценко Егор Андреевич**, студент групп БПИ193 факультета компьютерных наук НИУ ВШЭ.

8. ПОРЯДОК КОНТРОЛЯ И ПРИЁМКИ

Проверка программного продукта, в том числе и на соответствие техническому заданию, осуществляется исполнителем вместе с заказчиком согласно «Программе и методике испытаний», а также пункту 5.2.

Защита выполненного проекта осуществляется коммисией, состоящей из преподавателей департамента программной инженерии, в утверждённые приказом декана ФКН сроки.

9. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Botnet 2014 | Datasets | Research | Canadian Institute for Cybersecurity | UNB //

URL: https://www.unb.ca/cic/datasets/botnet.html (дата обращения: 13.05.2020);

2. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. // URL: https://lms.hse.ru/student.php?ctg=content&edit=322581&view=2067330 (необходима авторизация, дата обращения: 13.05.2020);

3. ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению. // URL: https://lms.hse.ru/student.php?ctg=content&edit=322581&view=2067331 (необходима авторизация, дата обращения: 13.05.2020);

4. ГОСТ 19.401-78 Текст программы. Требования к содержанию и оформлению. // URL: https://lms.hse.ru/student.php?ctg=content&edit=322581&view=2067332 (необходима авторизация, дата обращения: 13.05.2020);

5. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению. // URL: https://lms.hse.ru/student.php?ctg=content&edit=322581&view=2067334 (необходима авторизация, дата обращения: 13.05.2020);

6. ГОСТ 19.505-79 Руководство оператора. Требования к содержанию и оформлению. // URL: https://lms.hse.ru/student.php?ctg=content&edit=322581&view=2067334 (необходима авторизация, дата обращения: 13.05.2020);

7. ГОСТ 19.504-79 Руководство программиста. Требования к содержанию и оформлению. // URL: https://lms.hse.ru/student.php?ctg=content&edit=322581&view=2067337 (необходима авторизация, дата обращения: 13.05.2020).

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Изм.	Номера листов (страниц)				Всего листов (страниц) в документе	№ документа	Входящий № сопроводительного документа и дата	Подпись	Дата
	измененных	замененных	новых	аннулированных					